1

Exploiting Distinguishable Image Features in Robotic Mapping and Localization

Patric Jensfelt[1], John Folkesson[1], Danica Kragic[1] and Henrik I. Christensen[1]

Centre for Autonomous Systems
Royal Institute of Technology
SE-100 44 Stockholm, SWEDEN
[patric,johnf,danik,hic]@nada.kth.se

Summary. Simultaneous localization and mapping (SLAM) is an important research area in robotics. Lately, systems that use a single bearing-only sensors have received significant attention and the use of visual sensors have been strongly advocated. In this paper, we present a framework for 3D bearing only SLAM using a single camera. We concentrate on image feature selection in order to achieve precise localization and thus good reconstruction in 3D. In addition, we demonstrate how these features can be managed to provide real-time performance and fast matching to detect loop-closing situations. The proposed vision system has been combined with an extended Kalman Filter (EKF) based SLAM method. A number of experiments have been performed in indoor environments which demonstrate the validity and effectiveness of the approach. We also show how the SLAM generated map can be used for robot localization. The use of vision features which are distinguishable allows a straightforward solution to the "kidnapped-robot" scenario.

1.1 Introduction

It is widely recognized that an autonomous robot needs the ability to build maps of the environment using natural landmarks and to use them for localization, [2, 4, 18–20]. One of the current research topics related to SLAM is the use of vision as the only exteroceptive sensor, [3,5,6,15,17] due to the low cost. In this paper, we present a SLAM system that builds 3D maps using visual features using a single camera. We describe how we deal with a set of open research issues such as producing stable and well-localized landmarks, robust matching procedure, landmark reconstruction and detection of loop closing. These issues are of extreme importance when working for example in an EKF setting where the computational complexity grows quadratically with the number of features. Robust matching is required for most recursive formulations of SLAM where decisions are final.

Besides low cost, another aspect of using cameras for SLAM is the much greater richness of the sensor information as compared to that from, for example, a range sensor. Using a camera it is possible to recognize features based on their appearance. This can then simplify one of the most difficult problems in SLAM, namely data

association. We demonstrate just how powerful an advantage this is by using the SLAM map to perform robot localization without any initial pose estimate.

The main contributions of this work are i) a method for the initialization of visual landmarks for SLAM, ii) a robust and precise feature detector, iii) the management of the measurement to make on-line estimation possible, and iv) the demonstration of how this framework can facilitate loop closing and localization. Due to the low complexity, we believe that our approach scales to larger environments.

1.2 Related Work

Bearing only/single camera SLAM suffers from the problem that a single observation of a landmark does not provide an estimate of its full pose. This problem is typically addressed by combining the observations from multiple views as in the structure-from-motion (SFM) approaches in computer vision. The biggest difference between SLAM and SFM is that SFM considers mostly batch processing while SLAM typically requires on-line real-time performance.

The most important problem that has to be addressed in bearing only SLAM is landmark initialization, again because a single observation does not allow all degrees of freedom to be determined. A particle filter used to represent the unknown initial depth of features has been proposed in [3]. The drawback of the approach is that the initial distribution of particles has to cover all possible depth values for a landmark which makes it difficult to use when the number of detected features is large. A similar approach has been presented in [9] where the initial state is approximated using a Gaussian Sum Filter for which the computational load grows exponentially with number of landmarks. The work in [10] proposes an approximation with additive growth.

Similarly to our work, a multiple view approach has been presented in [6]. This work demonstrates the difficulties related to landmark reconstruction when the robot performs only translational motion along the optical axis. To cope with the reconstruction problem, a stereo based SLAM method was presented in [17] where Difference-of-Gaussians (DoG) is used to detect distinctive features which are then matched using SIFT descriptors. One of the important issues mentioned is that their particle filter based approach is inappropriate for large-scale and textured environments. The contribution of our work is that we deal with this problem using a feature detector that produces fewer features (presented in more detail in Section 1.4). In our work we use only a few high quality features for performing SLAM.

We have also considered another problem raised in [17] related to time consuming feature matching and use *KD*-trees to make our matching process very fast. The visual features used in our work are Harris corner features across different scales represented by a Laplacian pyramid for feature detection. For feature matching, we take a combination of a modified SIFT descriptor and *KD*-trees.

In man-made environments, there are many repetitive features and a single SIFT descriptor is not discriminative enough in itself to solve the data association problem. To deal with this problem, the approach using “chunks” of SIFT points to represent

landmarks in an outdoor environment has been presented in [12]. This is motivated by the success that SIFT has had in recognition applications where and object/scene is represented as a set of SIFT points. In our approach, the position of a landmark is defined by a series of single modified SIFT points representing different views of the landmark. Each such point is accompanied with a chunk of descriptors that make the matching/recognition of landmarks more robust. Our experimental evaluation shows also that our approach performs successful matching even with a narrow field of view which was mentioned as a problem in [6], [17].

One of the more challenging problem in SLAM is loop closing. In [15] visually salient so called "maximally stable extremal regions" or MSERs, encoded using SIFT descriptors, are used to detect when the robot is revisiting an area. A number of approaches have been presented for other sensory modalities, [7]. We also show how our framework can be used for this purpose.

In our work, a distinction is made between recognition and location features. A single location feature will be associated with several recognition features. The recognition features' descriptors then give robustness to the match between the location features in the map and the features in the current image. The key idea is to use a few high quality features to define the location of landmarks and then use the other features for recognition.

1.3 Landmark Initialization

In this work, we use feature matching across N frames to determine which points make good landmarks. Features that are successfully matched over a number of frames are candidates for landmarks in the map. Having a buffer of N frames also allows us to calculate an estimate of the 3D position of the corresponding landmark. The SLAM process is fed with data from the output side of the buffer, i.e. with the data from frames that are delayed N steps with respect to the input side of the buffer. Fig. 1.1 illustrates the idea.

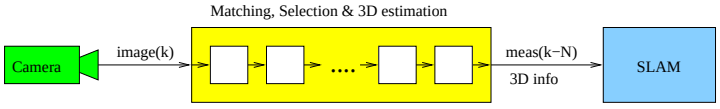

Fig. 1.1. Features are tracked over N frames in a buffer to allow for quality assessment and feature selection. The 3D position is estimated and landmarks formed from features that are tracked over many frames and are of high quality. The input to the SLAM process is delayed by N frames, but in return an estimate of the 3D position of the points landmarks can be supplied.

Since an estimate of the 3D position of landmarks can be supplied with the first observation of a landmark presented to the SLAM process (see Fig. 1.1), the landmarks can immediately be fully initialized in the SLAM process. This allows im-

mediate linearization without the need to apply multiple hypotheses [10] or particle filtering [3] techniques to estimate the depth. It is important to point out that that the approximate 3D position found from the buffer of frames is only used to be able to initialize the point landmark at the correct depth with respect to the camera at the first observation. In a sense it allows us to know which of the multiple hypotheses about the depth is correct right away which of course saves computations. Having the correct depth allows us, as said before, to handle the linearization errors that would results from having a completely wrong estimate of the depth.

In our experience having the SLAM output lagging N frames is not a problem. For cases where the current position of the robot is needed, such as when performing robot control, a prediction can be made using dead-reckoning forward in time from the robot pose given by the SLAM process. For typical values of N, the addition to the robot position error caused by the dead-reckoning is small and we believe that the benefits of being able to initialize landmarks using bearing-only information and perform feature quality checks are more significant. It is also important to understand that we predict forward the short distance from the SLAM time to the current time in every iteration so the errors from the prediction do not accumulate.

When selecting what features to use as landmarks in the map several criteria are considered: i) The feature is detected in more than a predefined number of frames, ii) The image positions of the feature allow good triangulation, and iii) The resulting 3D point is stable over time in the image. The first requirement removes the noise and moving targets. The second removes the features that have a high triangulation uncertainty (small baseline, features with bearings near the direction of motion). The third requirement removes features that lack sharp positions in all images due to parallax or a lack of a strong maximum in scale space. Difference in scales of the images can also cause apparent motion of features, such as for example a corner of a non-textured object.

In our implementation the length of the buffer, i.e. the number N is fixed. We have tested with values between 10 and 50. Since one of the key purposes with the buffer is to allow for 3D estimation we demand that the camera has moved to add a new frame to the buffer. This way, it is likely that there is a baseline for estimating the location. The value of N depends very much on the motion of the robot/camera and the camera parameters. For a narrow field of view camera mounted in the direction of motion of the robot as in our case the effective baseline will be quite small. As part of our future work we plan to compare the results that we get from using an omnidirectional camera.

1.4 Feature Description

It has been shown in [14] that the SIFT descriptor [11] is the most robust regarding scale and illumination changes. The original version of the SIFT descriptor uses feature points determined by the peaks of a series of Difference of Gaussians (DoG) on varying scales. In our system, peaks are found using Harris-Laplace features, [13] since they respond to regions of high curvature, instead of blob-like image structures

obtained by series of DoG. This leads to features accurately localized spatially, which is essential when features are used for reconstruction and localization, instead of just recognition. In addition, Harris-Laplace generates on average one fourth of points [8] compared to the regular DoG approach which is an important benefit for SLAM where we strive to keep the number of features low for computational reasons.

The original SIFT descriptor assigns canonical orientation at the peak of smoothed gradient histograms. This means that similar corners but with a significant rotation difference can have similar descriptors. In a sparse, indoor environment many of the detected features come from corner features and there may potentially be many false matches. For example, the four corners of the waste bin in Fig. 1.2 might all match if rotated. Therefore, we use a rotationally 'variant' SIFT descriptor where we avoid the canonical orientation at the peak of smoothed gradient histogram and leave the gradient histogram as it is.

Fig. 1.2. Example image from an indoor environment showing that many corner structures are very similar if rotated. Examples are the corners of the waste bin and the window sections on the door.

1.5 Feature Tracking

As outlined in Section 1.3 a buffer with the last N frames is stored in memory. The image data itself does not need to be stored, it is the feature points extracted from the frames that are stored. The feature points are tracked in consecutive image frames as the robot moves. The similarity between two feature points is the distance between the descriptors which are vectors in a 128-dimensional space. To manage the matching between frames, lists with association/points are maintained. Fig. 1.3 shows the basic organization of this frame memory. On the left we have the buffer with the N frames drawn vertically with the input side at the top and output side that is fed into the SLAM process at the bottom. On the right side of the buffer in the frame memory we show the association list that keep track of which feature points in the

different frames have been matched. Each such association list item corresponds to one landmark in the world.

The SIFT descriptor of a feature changes when the camera moves. However given that the feature is continuously tracked this change in descriptor is typically small between consecutive frames and can be handled. Each association/point in the world, p_i, will have a set of descriptors d_j corresponding to different vantage points for each of the tracked features over the N frames. These lists can be analyzed to judge the quality of the corresponding landmark candidate as described in Section 1.3. The output from the tracking module is a selection of the points in the oldest frame in the buffer. These points correspond to either already initialized landmarks or new landmarks of high quality that can be initialized. Along with new landmarks an estimate of the 3D position is provided. This estimate is only used in the initialization step and thus only have to be performed for new landmarks. The number of points that are output is only a small fraction of all points detected in a frame.

To make the matching procedure faster and more robust, we predict the approximate image location of the features from previous frames using odometry and optical flow estimates. The buffer allows us to predict the position of features detected not only in the previous frame but also in frames before that. We make one prediction for each potential landmark in the old frames, i.e., we only use the last of features that have been matched between frames. One important observation in our investigation was that even with very small changes between frames the same features would typically not be detected in every frame which would mean loosing track of most features if only matching against the previous frame. This observation is true also when using the DoG detector for the features.

Using the prediction allows us to reduce the search window when looking for matches between frames. Features that do not match the predicted positions of points currently in the frame memory are matched to landmarks in the database. This also allows for fast detection of potential loop closing situations. The first time the location of a landmark has been established through triangulation and it has met the other criteria listed in Section 1.3, it is added to the database as a new location feature. This is discussed further in the next section.

1.6 Database Management

As the robot moves along in the environment, features will leave its field of view and thus eventually also the frame memory described in Section 1.5. In SLAM, it is important for the robot to detect if it is revisiting an area and find the correspondence between the new features in the image and existing landmarks. In fact, this is an issue not only when revisiting an area after closing a large loop but also when turning the camera back to an area that has not been looked at for a while. To handle this we use a database that stores information about the appearance of the landmarks from different view points. That is, we let the SLAM process take care of estimating the position of the landmarks and leave it to the data base to deal with the appearance and matching of them.

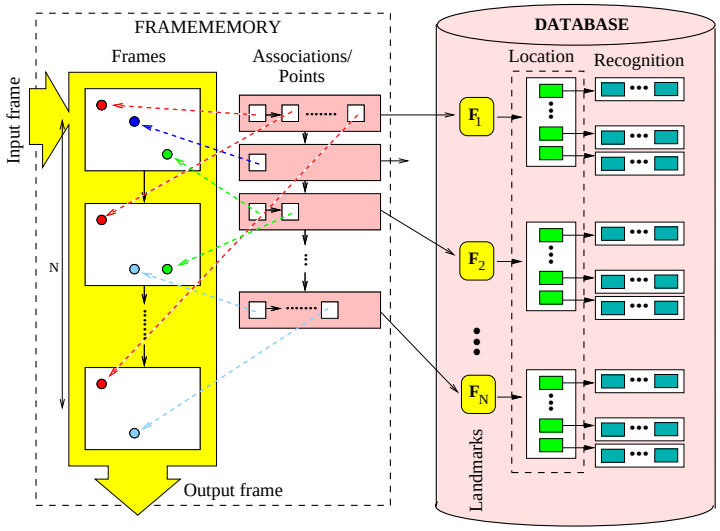

Fig. 1.3. A schematic view of the frame memory that stores the points and the associations between points in the N last frames and the database. The point in the frame memory that have been matched or added to the database have a reference to the corresponding landmark there. Each landmark in the database has a set of descriptors that corresponds to location features seen from different vantage points. To validate a match, each of these descriptors keeps a list of the other descriptors found in the same frame. We refer to these as recognition descriptors. These provide the ability to "recognize" a landmark again.

In our database, each landmark has a set of SIFT descriptors that correspond to different vantage points. These descriptors are provided by the frame memory that matches the image points between frames and stores this in the association/point lists. A new descriptor is added to a landmark when it differs more than a certain threshold from all other descriptors attached to that landmark. Fig. 1.3 shows the structure of the database where the landmarks are denoted with $F_1, F_2, \ldots, F_N$. The dashed box contains the descriptors for each of the landmarks. A KD-tree representation and a Best-Bin-First [1] search allow for real-time matching between new image feature descriptors and those in the database.

There may be potentially many similar descriptors corresponding to different features. Looking once again at the image in Fig. 1.2 the four corresponding corner points on the glass windows on the left hand side door section will all be very similar. Trying to find a correspondence between images features in a new frame and the map landmarks with a bit of uncertainty induced by a loop can be hard based only on a single SIFT descriptor. Therefore, we require multiple matches to increase the robustness [6]. We refer to these multiple descriptors as recognition descriptors and the corresponding image features as recognition features. That is, when matching

a feature to the database we first look for matches between its location descriptor[1] and the descriptors in the database. Then, we verify the match using the corresponding two sets of recognition descriptors. As an additional test, it is required that the displacement in image coordinates for the descriptor is consistent with the transformation between the two frames estimated from the matched recognition descriptors. Currently, the calculation is simplified by checking the 2D image point displacement. This final confirmation eliminates matches that are close in the environment and thus share recognition descriptors such as would be the case with the glass windows in Fig. 1.2.

To summarize, the matching of new features with the database proceeds as follows:

1. Find matching candidates by matching with the set of location descriptors in the database (dashed box in the database in Fig. 1.3). The KD-tree representation allows for fast matching and an effective way to eliminate all but a few of the potential location feature matches in the database.
2. Validate the candidates using all extracted descriptors from the current frame, i.e. the recognition feature and the recognition features associated with the matches from step 1.
3. Confirm candidate by checking that the motion given all the matches is consistent.

1.7 Using the Location Features for SLAM and Localization

We have seen in the previous sections how features are tracked between frames using a buffer of N frames and how keeping this buffer allows for judging the quality of potential landmarks and for finding an estimate of the 3D position of the landmarks before they are initialized in SLAM. We use an EKF based implementation for SLAM but the output from the frame memory and database can be fed into any number of SLAM algorithms. The main difference between the EKF implementation used here for SLAM and the standard formulation is that we supplement the first bearing-only measurement of a new landmark with the additional information about the approximate distance as determined through the triangulation in the frame buffer. The distance information s not given in the form of an EKF measurement, these are always only bearings. It is used more like some oracle told the SLAM filter where in depth to initialize a new landmark. As a result an accurate linearization can be made in the EKF as of the first sighting without the need for any special arrangement to account for an unknown depth as in e.g. [3, 10].

Once the map is built, it can be used for localization. Since the landmarks in the database are distinguishable, it allows the robot to recognize areas that are part of its map using a single landmark. Thus the robot can automatically initialize itself

[1] Each feature in a new image is a potential location feature and the rest of the features in that frame will be its recognition features. A feature can thus be both a location feature and act as recognition feature for one or more other location features in that frame.

in the map after which the map can be used to track the robot pose. This is similar to the situation when closing a loop and having to make the association between new measurements and old landmarks in the map. In the latter case one has a rough idea about where to look for such a match. When performing global localization the uncertainty is the entire space and the search for matches can thus become quite expensive. The framework presented in this work allows for fast matching against the database which we will further investigate in the experimental section.

One of the benefits of our representation with multiple descriptors for the landmarks, each of them encoding the appearance from a certain vantage point, is that this information can be used to deduce the approximate position of the robot from a single point observation. To allow for this, the pose from which each descriptor was first observed is stored along with the descriptor. The idea is not that to get an exact position out of an observation but to narrow down the area that the robot is likely to be in.

To find the pose of the robot in the map we initialize a particle filter as soon as a match to the database is found. The particles are initially distributed around the pose indicated by the database match. The orientation for each particle in the initial sample set is given by the bearing angle of the observation. This cuts the unconstrained degrees of freedom down to two. Given that the 3D position of the landmark is given, that the floor is flat and that a measurement of the bearing to the landmark is given not only in the xy-plane but also in the vertical direction the distance to the landmark can also be estimated from a single measurement. In our current implementation have not incorporated this last bit of information. After the first observation it will then require only a few other database matches to reduce the spread of particles enough to say that the position is known. In our current implementation we initialize a normal EKF localization algorithm at this point.

1.8 Experimental Evaluation

The experimental evaluation has been carried out on a PowerBot platform from ActivMedia. It has a differential drive base with two rear caster wheels. The camera used in the experiments is a Canon VC-C4 CCD camera. This camera has built-in pan-tilt-zoom capabilities but in the experiments these were all fixed with a slight tilt upwards to reduce the amount of floor in the image. The non-holonomic motion constraints of the base makes it hard to generate large baselines for triangulation as the motion mostly is along the optical axis. This in combination with the relatively narrow field of view[2] contributes to the difficulty of the problem.

The experimental evaluation is carried out in two steps. In the first step we let the robot build up a database of visual landmarks while feeding them into SLAM to build a map with the 3D position of these landmarks. This test will show that i) our framework produces few but stable landmarks well suited for map building ii) we can match new observation to the database when closing the loop.

[2] The field of view of the Canon VC-C4 is about 45° in the horizontal plane and 35° vertically when the camera is in "wide-angle mode" as in our experiments.

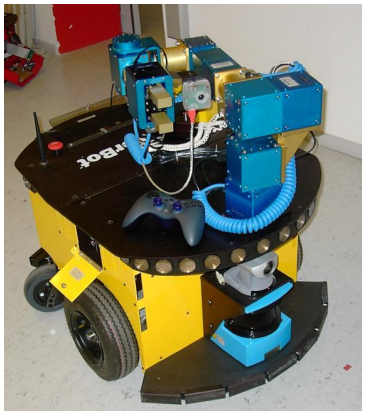

Fig. 1.4. The experimental PowerBot platform with the Canon VC-C4 camera mounted on the laser scanner.

In the second step the robot is given the task to find its position in this map given the database and the map. This test will underscore the ability to match observations to the database without false matches and highlight the strength our representation provides to a localization system. The setting for the experiments is an area around an atrium that consists of loops of varying sizes.

In the map building experiment we let the robot drive 3 laps around a part of the environment, each lap being approximately 30m long. This trajectory is shown along with a map built using a laser scanner in Fig. 1.5 (dark "circular" path). The experiment took 8 minutes and 40 seconds and the time to process the data was 7 min 7s on a 1.8GHz laptop computer which shows that the system can run in real-time even if all unmatched features are matched to the database in every processed frame. After the first loop the database/map consisted of 98 landmarks. The landmarks are shown as red/dark squares in Fig. 1.5. After the 3 loops were completed the map consisted of 113 landmarks. This shows that the robot was successfully matching most of the observations to the database during the last two laps. Otherwise one would expect the map to be roughly 3 times the size after the first loop.

There are two important characteristics to note about the map with 3D visual landmarks. Firstly, there are far fewer features then is typically seen in other works where SIFT like features are used in the map [16, 17]. This can be attributed to using only the most stable points features as SLAM landmarks and the rest for recognition/matching of those landmarks. Real-time performance was not demonstrated in [16, 17].

Secondly, the landmarks are well localized which can be seen by comparing with the walls from the superimposed map built using the laser scanner[3]. Some of the landmarks are lamps hanging from the ceiling such as the one at the upper right

[3] Note that the laser based map is only shown for reference. Only vision was used in the experiments

corner of the robot path in Fig. 1.5. This lamp is also visible in the upper image on the right side of the same figure. The area to the right in the map with a large number of landmarks is shown in the lower right image in Fig. 1.5. This area has many objects at differing depths. In the back against the walls there are five heavily textured paintings. The line in the laser based map comes from the benches that are in front of the wall which accounts for the line of visual landmarks behind the line made using the laser. Part of the spread is also due to uncertainty in depths of the landmarks. The robot moved close to orthogonal to the wall creating very little baseline in the data fed into SLAM. It is worth while repeating that the depth estimate from the frame memory only serves to get the initialization of the depth roughly right to reduce the linearization errors but that the real estimate of the 3D position is calculate through the SLAM process using the bearing-only measurements.

Fig. 1.5. The SLAM map is shown as red squares at the locations of landmarks. To help visualize the environment, a separately made laser scanner map is superimposed on this map. Also shown are the trajectories from when the map was built (3 loops) and when the robot was trying to localize. The feature match between the evening database image image (top) and the daytime localization experiment (middle) is indicated. The bottom image corresponds to the area of the map where the density of landmarks is greatest.

Depending on the scene, a typical frame has between 40-100 point features. The time to perform the tracking over frames has constant complexity. Out of the features

in each new frame as many as half typically do not match any of the old features in the frame memory and are thus matched to the database. The matching to the database uses the KD-tree in the first step which makes this first step fast. This often results only in a few possible matching candidates. A typical landmark in the database has around 10 descriptors acquired from different viewing angles.

As we noted while building this map, the landmarks inserted into the database during the first loop were then matched and updated on the second two loops with no difficulty. In order to demonstrate the usefulness of this result we then used the map and database in the second experiment to localize the robot at a different time of day, when there was sunlight streaming through the windows and students milling about. When performing localization we do not need to estimate the 3D location of the points and thus the output from the frame memory is shifted to be at the input side, i.e., without the delay. The frame memory is still used to track the points over time but not to verify the quality which is most important when building the map. The robot was given no initial pose information relative to the database and map. In Fig. 1.5 the path of the robot in this second experiment is overlayed. The robot initially moves around in the lower left corner of the figure before it moves to the right for some time and then eventually moves up into regions that were mapped in the first experiment. The robot moves in unmapped areas for almost 6 minutes. During this time it is constantly trying to find matches between unmatched features in the image to the database. More than 1,000 images are added to the frame memory and matched without any false matches to the database. When the robot enters an area were it can observe landmarks from the database/map it almost immediately recognized a landmark and a particle filter consistent with the observation was initialized. In seconds, after repeated observations of several landmarks, the particles had converged enough to initialize the EKF localization. The evolution of the particle filter from initialization to convergence is shown in Fig. 1.6 in four frames from left to right. In the last frame in this figure, the precision of the localization is indicated by the alignment of the laser scan (not used for localization, only for comparison) and the walls in the map.

The problems of a narrow field of view for an ordinary camera were overcome in these experiments partly by carefully driving the robot. Better results were observed when the robot was driven so that features remained in view for as long as possible, (notice the rather crooked paths in Fig. 1.5).

1.9 Conclusion and Future Work

The contributions of the framework presented here are the feature selection and matching that allows for real-time vision based bearing-only SLAM. We distinguish between location and recognition features. The location features correspond to points in 3D for which the robot motion allows good triangulation and which are used as landmarks in the map. The matching is made robust by the inclusion of many recognition features from the image for each location feature.

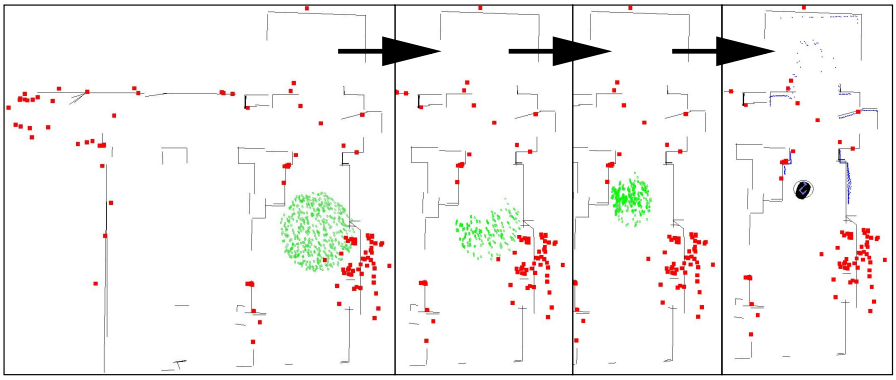

Fig. 1.6. Here we show the evolution of the localization particle filter. On the left we see the initial distribution after first match to the database. On the right is shown the robot after initialization in the map. The laser scan is also shown in the right image to help confirm that the localization is indeed correct.

The use of Harris-Laplace corner detection combined with a scale space maximization gives rotationally variant features which are more appropriate for the camera motions generated by a camera mounted on a mobile robot.

We use three criteria to select features for SLAM: persistence, range estimate quality and image stability. The persistence criteria eliminates spurious and dynamic features by requiring that the feature be observed many times. The quality of the range estimate depends on the motion of the robot relative to the 3D point. If the motion does not produce a sufficient baseline there is no reason to use the associated vision feature. The stability in the image depends on how well the feature is localized in the image.

The evaluation of our vision based landmarks was done on data collected from a robot moving through a indoor environments. We were able to report no false matches and the creation of accurate 3D maps. We also showed that those maps could be used to localize the robot automatically in the environment.

One topic for future research is to actively control the pan-tilt degrees of freedom of the camera on the robot. This would allow the robot to focus on a landmark for a longer time and create a better baseline and thus a more accurate map. Another way to address the problem with the limited field of view is to use an omnidirectional camera which is also part of the planned future work.

References

1. J. S. Beis and D. G. Lowe. Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. In *IEEE CVPR*, pages 1000–1006, 1997.
2. J. A. Castellanos and J. D. Tardós. *Mobile Robot Localization and Map Building: A Multisensor Fusion Approach*. Kluwer Academic Publishers, 1999.

3. A. J. Davison. Real-time simultaneous localisation and mapping with a single camera. In *ICCV*, 2003.
4. G. Dissanayake, P. Newman, S. Clark, H.F. Durrant-Whyte, and M. Corba. A solution to the slam building problem. *IEEE TRA*, 17(3):229–241, 2001.
5. J. Folkesson, Jensfelt P, and H. I. Christensen. Vision slam in the measurement subspace. In *IEEE ICRA05*, 2005.
6. L. Goncavles, E. di Bernardo, D. Benson, M. Svedman, J. Ostrovski, N. Karlsson, and P. Pirjanian. A visual fron-end for simultaneous localization and mapping. In *IEEE ICRA*, pages 44–49, 2005.
7. J. Gutmann and K. Konolige. Incremental mapping of large cyclic environments. In *IEEE Int. Symposium on Computational Intelligence in Robotics and Automation*, volume 1, pages 318–325, 1999.
8. Patric Jensfelt, Danica Kragic, John Folkesson, and Mårten Björkman. A framework for vision based bearing only 3D SLAM. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA'06)*, Orlando, FL, May 2006.
9. N. M. Kwok, G. Dissanayake, and Q.P. Ha. Bearing only SLAM using a SPRT based gaussian sum filter. In *IEEE ICRA05*, 2005.
10. T. Lemaire, S. Lacroix, and J. Solà. A practical 3D bearing-only SLAM algorithm. In *IEEE/RSJ IROS*, pages 2757–2762, 2005.
11. David G. Lowe. Object recognition from local scale-invariant features. In *ICCV*, pages 1150–57, 1999.
12. R.H. Luke, J. M. Keller, M. Skubic, and S. Senger. Acquiring and maintaining abstract landmark chunks for cognitive robot navigation. In *IEEE/RSJ IROS*, 2005.
13. K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. In *IEEE ICCV*, pages 525–531, 2001.
14. K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. In *IEEE CVPR*, pages 257–263, 2003.
15. P. Newman and K. Ho. SLAM-loop closing with visually salient features. In *IEEE ICRA*, pages 644–651, 2005.
16. S. Se, D. G. Lowe, and J. Little. Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks. *IJRR*, 21(8):735–58, 2002.
17. R. Sim, P. Elinas, M. Griffin, and J. J. Little. Vision-based slam using the rao-blackwellised particle filter. In *IJCAI Workshop on Reasoning with Uncertainty in Robotics*, July 2005.
18. J.D. Tardós, J. Neira, P.M. Newman, and J.J. Leonard. Robust mapping and localization in indoor environments using sonar data. *IJRR*, 4, 2002.
19. S. Thrun, D. Fox., and W. Burgard. A probalistic approach to concurrent mapping and localization for mobile robots. *Autonomous Robots*, 5:253–271, 1998.
20. S. Thrun, Y. Liu, D.Koller, A. Ng, Z. Ghahramani, and H. Durrant-White. SLAM with sparse extended information filters. *IJRR*, 23(8):690–717, 2004.